What is claimed is:

1. A multi-threading processor, comprising:

a first instruction fetch unit and a second instruction fetch unit;

a multi-thread scheduler unit coupled to said first instruction fetch unit and said second instruction fetch unit;

an execution unit coupled to said scheduler unit, wherein said execution unit is to execute a first active thread and a second active thread; and

a register file coupled to said execution unit, wherein said register file is to switch one of said first active thread and said second active thread with a first inactive thread.

2. A multi-threading processor as recited in claim 1, further comprising an on deck context unit coupled to the register file, wherein said on deck context unit is to maintain a first inactive thread and a second inactive thread.

3. A multi-threading processor as recited in claim 2, wherein said register file is to switch one of the first active thread and the second active thread with a second inactive thread.

4. A multi-threading processor as recited in claim 3, further comprising:

a first instruction decode unit coupled between the first instruction fetch unit and the scheduler unit; and

a second instruction decode unit coupled between the second instruction fetch unit and the scheduler unit.

5. A multi-threading processor as recited in claim 1, wherein the scheduler unit is a four thread scheduler unit, further comprising:

a third instruction fetch unit coupled to said four thread scheduler unit; and

a fourth instruction fetch unit coupled to said four thread scheduler unit.

6. A multi-threading processor as recited in claim 5, wherein said register file is a four way register file.

7. A multi-threading processor as recited in claim 6, wherein said register file is to switch one of the first active thread and the second active thread with a second inactive thread.

8. A multi-threading processor as recited in claim 7, further comprising:

a third instruction decode unit coupled between the third instruction fetch unit and the four thread scheduler unit; and

a fourth instruction decode unit coupled between the fourth instruction fetch unit and the four thread scheduler unit.

9. A method for switching threads in a multi-threading processor, comprising:

fetching a first active thread and a active second thread;

detecting a stalling event in said first active thread; and

switching said first active thread with a first inactive thread, if said first inactive thread is ready to execute.

10. A method for switching threads as recited in claim 9, further comprising executing said first inactive thread and said second active thread.

11. A method for switching threads as recited in claim 9, further comprising detecting a stalling event in the second active thread.

12. A method for switching threads as recited in claim 11, further comprising switching said second active thread with a second inactive thread, if said second inactive thread is ready to execute.

13. A method for switching threads as recited in claim 12, further comprising executing the first inactive thread and the second inactive thread.

14. A method for switching threads as recited in claim 9, further comprising executing the second active thread, if the first inactive thread is not ready to execute.

15. A set of instructions residing in a storage medium, said set of instructions capable of being executed by a processor for searching data stored in a mass storage device comprising:

fetching a first active thread and a active second thread;

detecting a stalling event in said first active thread; and

switching said first active thread with a first inactive thread, if said first inactive thread is ready to execute.

16. A method for switching threads as recited in claim 15, further comprising executing said first inactive thread and said second active thread.

17. A method for switching threads as recited in claim 15, further comprising detecting a stalling event in the second active thread.

18. A method for switching threads as recited in claim 17, further comprising switching said second active thread with a second inactive thread, if said second inactive thread is ready to execute.

19. A method for switching threads as recited in claim 18, further comprising executing the first inactive thread and the second inactive thread.

20. A method for switching threads as recited in claim 15, further comprising executing the second active thread, if the first inactive thread is not ready to execute.